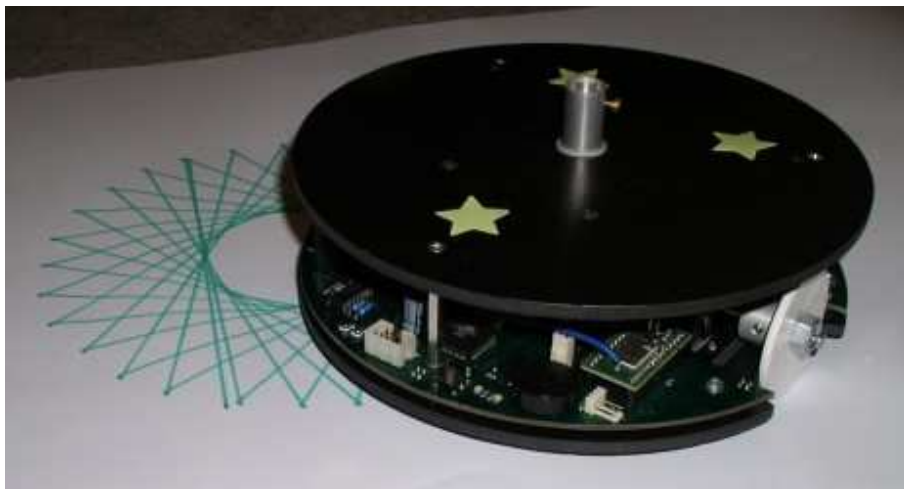


## Cvičenie: Mobilná robotika



Obr. 24: Robotnačka – kresiaci mobilný robot.

### Robotnačka

Robotnačka je autonómny mobilný robot, ktorý vie kresliť. Určený je predovšetkým pre kreslenie pomocou programovacieho jazyka Imagine Logo. Slúži ako pomôcka pri výučbe základov programovania pre deti. Robot vyvinula skupina nadšencov v rámci projektu financovaného firmou MicroStep-MIS v rokoch 2001 (prvý model) až 2005 (súčasná verzia). Dnes je to hotový produkt, pripravený na experimentálne nasadenie do vybraných škôl.

Robotnačka je postavená na kruhovej podstave z pevného plastu s priemerom 210 mm. Na nej sú umiestnené dva dvojfázové krokové motory Microcon SX 17, a voľné oporné koliesko (angl. *caster*) kvôli stabilite. Na základni je upevnená aj riadiaca elektronika vrátane vstavanej nabíjačky a olovený bezúdržbový akumulátor.

Robot po zapnutí nevykonáva žiadnu samostatnú činnosť, len čaká na povely, ktoré dostáva od užívateľa z nadradeného počítača. Robot je s počítačom spojený cez virtuálny sériový port rádiovým Bluetooth (BT) spojením, zabudovaný softvér prijíma a vykonáva navigačné príkazy (vpred, vzad, stoj,...) prijaté z éteru.

Na otočenie o 360 stupňov motory musia urobiť 2880 krokov a dĺžka dráhy zodpovedajúcej najmenšiemu kroku je približne 0,2 mm.

### Programovanie

Nasledujúci obrázok znázorňuje viacero možností ovládania a programovania robota. Ako sme už spomenuli, vstavaný software (firmware) robota čaká na príkazy, ktoré posielame cez Bluetooth rozhranie. Robot obsahuje jednoduchý interpret príkazov, ktoré sú vykonávané bezprostredne po ich

#### Technické parametre

Rozmery:

- priemer 210 mm,
- výška 70 mm,
- hmotnosť 1,9 kg (z toho batéria 0,7 kg),
- rozchod 180 mm,
- priemer kolieska 50 mm.

Pohon:

- 2 dvojfázové krokové motory Microcon SX 17-0402-09 s veľkosťou kroku 0,9° stupňa.
- 1 solenoid na ovládanie pera.

Riadiaca jednotka:

Mikroprocesor Atmel AT89S8252 (8 bit, jadro typu x51),  $f = 11,059 \text{ MHz}$ . 8kB FLASH, 2kB EEPROM, 256B RAM.

Akumulátor:

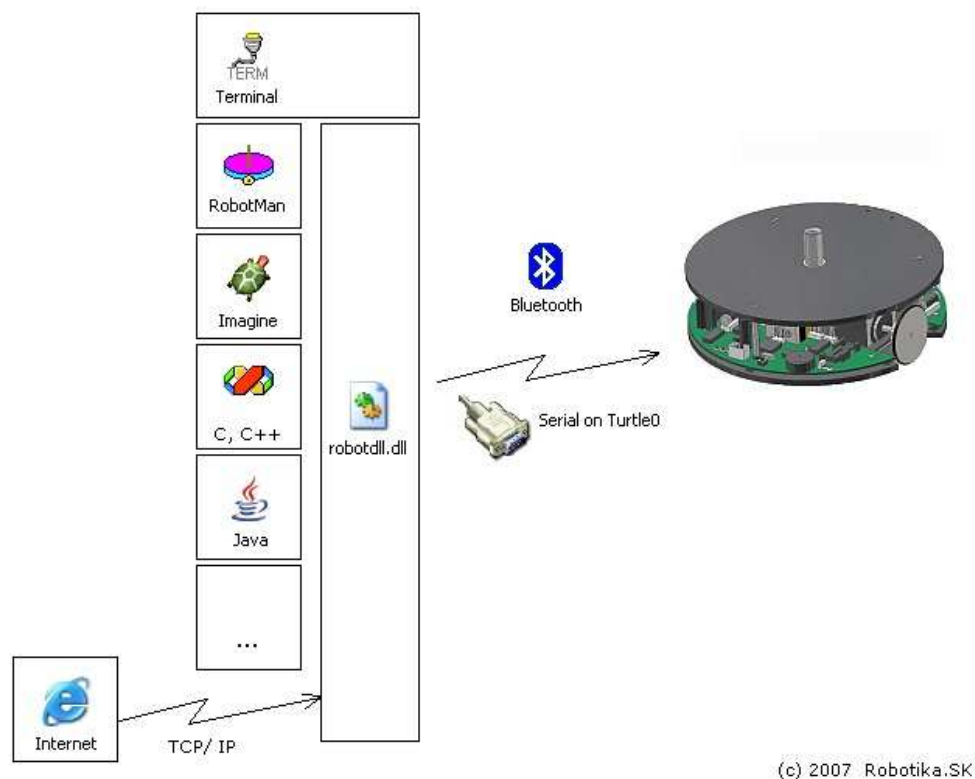
gélový olovený bezúdržbový Energuard CP632 (alebo podobný) 6 V / 3,2 Ah.

Komunikačné rozhranie:

bezdrôtové (Bluetooth) cez virtuálny sériový kanál. Pôvodný komunikačný modul vyvinutý firmou MicroStep-MIS založený na čipe Infineon ROK 104001121 (v súčasnosti sa už nevyrába).

Robot obsahuje jednoduchý interpret príkazov, ktoré sú vykonávané bezprostredne po ich

prijatí. Na každý úspešne prijatý a dekódovaný príkaz robot pošle aj odpoveď. Robota teda možno ovládať prostredníctvom ľubovoľného programu, ktorý dokáže posilať príkazy cez sériový port. Komunikačný protokol je zverejnený na stránkach robotika.sk v CVS projekte Robotnačka.



Obr. 25: Možnosti programovania Robotnačky.

Komfortnejší spôsob programovania robota spočíva vo využití knižnice `robotdll.dll`, ktorá okrem funkcií pre pohyb dokáže aj nadviazať spojenie s robotom. Funkcie tejto knižnice je možné volať z ľubovoľného jazyka, vrátane jazykov C, C++, Java a Imagine Logo.

Ďalej ukážeme manuálne ovládanie a programovanie v jazyku C. Ostatné spôsoby sú popísané na stránkach robotika.sk.

## Robotman

Pre interaktívnu prácu s Robotnačkou je najjednoduchšie použiť aplikáciu Robotman, ktorá umožní posielanie základných príkazov. Po zapnutí robota a jeho pripojení na virtuálny sériový port BT rozhrania možno priamo posilať príkazy. Tie sú rozdelené do troch skupín. V prvej skupina sú tzv. polohové príkazy – robot vykoná daný počet krokov a ostane stáť.

V druhej skupina sú tzv. rýchlostné príkazy, ktoré spôsobia zapnutie otáčanie motorov danou rýchlosťou až kým nepríde ďalší povel. Posledná skupina špeciálnych príkazov sa používa pre prácu so senzormi, pri diagnostike, alebo pri ovládaní kresliaceho pera.

Po spustení programu je potrebné do prvého riadku zadať číslo virtuálneho portu, ktorý sa vytvoril po pripojení robota cez Bluetooth rozhranie. Ak pripojenie prebehne úspešne, vedľa tlačidla sa zjaví nápis OK.

Potom je možné robota používať. Ak zadáte v sekcii polohových príkazov nejaké číslo a stlačíte tlačidlo pre niektorý smer, krokové motory robota vykonajú príslušný počet krokov v danom smere. Pod zmenami smeru sa rozumejú smery otáčania koliesok.

Podobne fungujú aj rýchlostné príkazy, je potrebné zadať dve čísla, ktoré znamenajú rýchlosti otáčania ľavého, resp. pravého kolieska. Rýchlosti  $x$  môžu byť celé čísla od 0 po 99, pričom vzťah so skutočnou rýchlosťou  $\omega$  je nasledovný:

$$v = \frac{1000x}{256 * 0,25 * 800} = 0,01953125x \quad [\text{ot/s}]$$

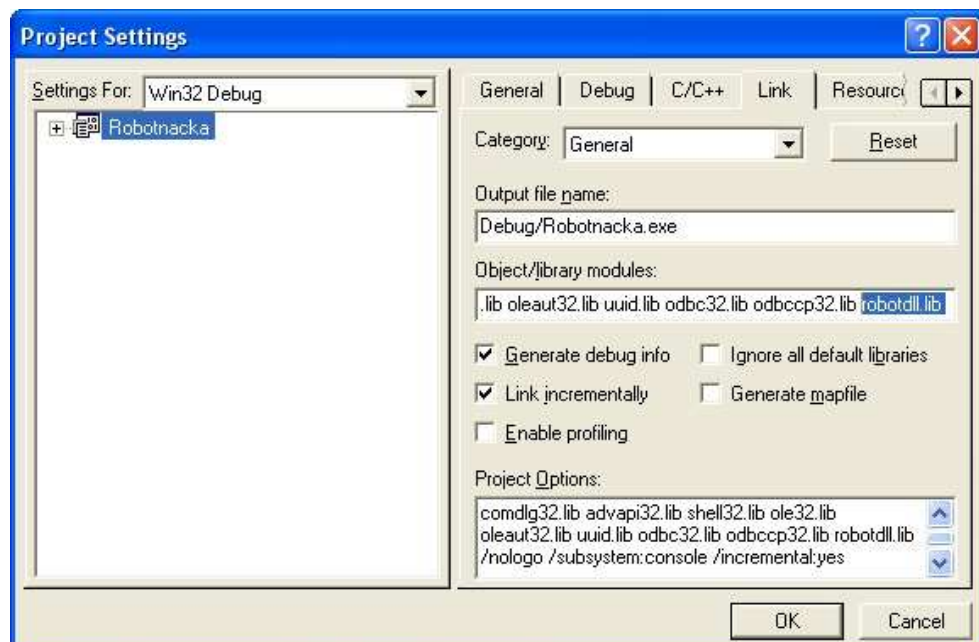
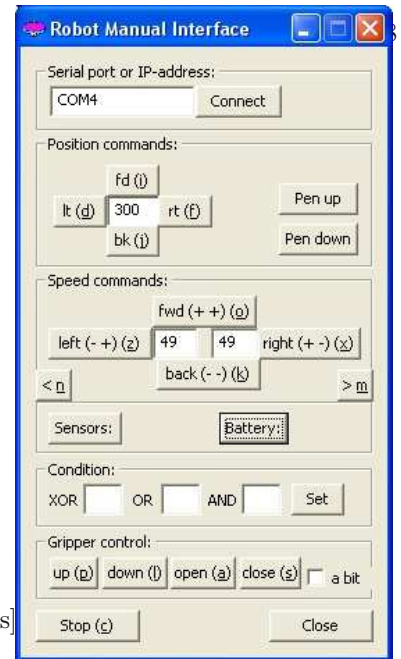
Priemerná doba trvania jedného kroku [ms] =  $(256/x) * 0,25$ , kde  $x$  je zadavane rýchlosť od 0 do 99. To vychádza ze pri maximalnej rýchlosti (800 krokov) je doba 517 ms, cize rýchlosť =  $1/0,517 = 1,94$  otacky za sekundu.

Cize výsledná rýchlosť otáčania kolesa [otacka/sekundu] =  $(1000 * x) / (256 * 0,25 * 800)$ , kde  $x$  je zadavane rýchlosť od 0 do 99.

### Programovanie v jazyku C

Nasledovný program predstavuje minimálny program, ktorý po inicializácii prihlási užívateľa a otvorí príslušný sériový port. Po úspešnom prihlásení robot urobí pohyb vpred a zatočí vpravo. V programe nie sú ošetrované chyby, automaticky sa (nesprávne) predpokladá, že všetko bude fungovať.

Pri preklade vo VisualC++ treba prilinkovať aj knižnicu `robotdll.dll`, teda v nastaveniach projektu (Project/Settings/Link) pridať na koniec zoznamu **Object/library modules:** aj `robotdll.dll` (viď obr.??).



Obr. 26: Nastavenie vo Visual C++.

Základné príkazy, ktoré máte k dispozícii a ktoré treba na cvičenia sú `robot_fd(r,kroky)`; `robot_rt(r,kroky)`; . Ostatné príkazy nájdete v dokumentácii.??

```
#include "robotdll.h"

int main(int argc, char *argv[])
{
    int r = robot_init(1);           // Initialize
    robot_user(r,"john","turtle");  // Set User / Password
    robot_connect(r, "com5");        // Open Connection
    robot_alwayswait(r,"on");        // Wait for finished commands

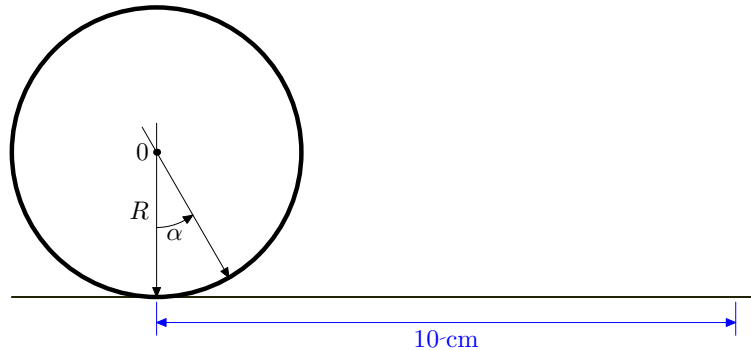
    for (int i=0; i<5; i++)
    {
        robot_fd(r,500);             // Go Forward
        robot_rt(r,1152);            // Rotate right
    }

    robot_done(r);                   // Close Connection
    return 0;
}
```

## Úloha 1: Prvé kroky

Vypočítajte a overte, koľko krokov musí robot urobiť, aby nakreslil rovnú čiaru dĺžky 10 cm.

Pozri video, prípadne simuláciu.



Obr. 27: Koliesko po rovnej dráhe.

Robot sa bude pohybovať po priamke vtedy, keď sa obe kolieska budú otáčať súčasne rovnakým smerom. Keďže viete, že Robotnačka je poháňaná krokovými motormi, viete z katalógového listu motora určiť, koľko krokov je potrebných na jednu úplnú otáčku. Keď si posuvným meradlom zmeriate priemer kolieska, budete vedieť aj to, akú dráhu pri tejto úplnej otáčke Robotnačka prejde. No a potom už bude hračka vypočítať počet impulzov, aby dráha bola 10cm.

polomer  $R = 25 \text{ mm}$

krok KM  $\alpha = 0,9^\circ = 2\pi 0,9/360 \text{ rad}$

$$s = N \cdot R \cdot \alpha \quad [\text{mm}; -, \text{mm}, \text{rad}]$$

$$N = \frac{s}{R \cdot \alpha} = \frac{100.360}{25.2.\pi.0,9} = \underline{\underline{509}}$$

Vypočítanú hodnotu overte. Podarilo sa?

**Pozor:** Všimnite si závislosť na rozmeroch kolieska. V prípade zmeny rozmerov, nepresného zmerania, starnutia materiálu a pod. sa chyba zväčší.

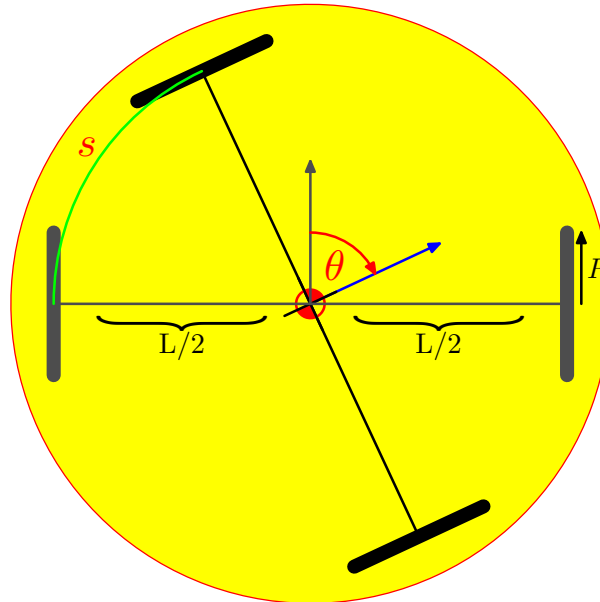
### Otázky:

Dokážete úlohu splniť presne? Aká je teoretická chyba v mm? (pomôcka: vyjde vám pri delení celé číslo?)

Aký je vplyv povrchu na riešenie úlohy? Čo sa stane, ak koliesko bude prešmykovať? Overte to tak, že nakreslíte niekoľkokrát za sebou štvorec a zistíte, ako sa postupne zväčšuje odchýlka od koncového bodu.

## Úloha 2: Vpravo vbok!

Vypočítajte a overte, koľko krokov musí robot urobiť, aby sa otočil na mieste o  $90^\circ$  v smere hodinových ručičiek. V prípade, že sa kolieska točia navzájom opačne ale s rovnakou rýchlosťou, bude sa robot otáčať okolo svojej osi. Pritom prejdú obe kolieska rovnakú dráhu, bude to časť kružnice, ktorej dĺžka závisí od počtu krokov. Využite výsledky predošlej úlohy a spočítajte, koľko krokov treba na otočenie o  $90$  stupňov.



Obr. 28: Koliesko po kruhovej dráhe.

Rozchod  $L = 180$  mm

Obe kolieska musia prejsť pri otáčaní rovnakú dráhu  $s$

$$s = \frac{L}{2}\theta \quad [\text{mm}; \text{mm}, \text{rad}]$$

Pre pohyb kolieska v závislosti na počte impulzov  $N$  zároveň platí podľa predošlej úlohy

$$s = N \cdot R \cdot \alpha \quad [\text{mm}; -, \text{mm}, \text{rad}]$$

Ak zlúčime obe rovnosti, dostaneme pre  $N$

$$N = \frac{\theta L}{2R\alpha}$$

(Pozor,  $\alpha$  aj  $\theta$  by mali byť v radiánoch, ale keďže prípadný konverzný faktor sa vykrátí, môžu byť aj v stupňoch, ale vždy oba uhly v rovnakých jednotkách!)

Pre požadované otočenie  $90^\circ$

$$N = \frac{90 \cdot 180}{2 \cdot 25 \cdot 0,9} = \underline{\underline{720}}$$

Vypočítanú hodnotu overte. Podarilo sa?

**Pozor:** Nemiešajte pri výpočtoch uhlové miery. Buď počítajte v stupňoch, alebo radiánoch, nie v oboch naraz!

### Otázky:

Ako vplývajú poruchové veličiny o ktorých sme sa zmienili pri riešení prvej úlohy na otáčanie robota? Aká je chyba natočenia, ak pri otáčaní o  $90^\circ$  vypadne jeden krok (napríklad pri príliš veľkom zrýchlení)?

V oboch úlohách sme uvažovali len o počte krokov. Neuvažovali sme o rýchlosti. Aký má vplyv na výsledok?

### Úloha 3: Točí sa mi točí...

Navrhnete vhodné rýchlosti pre obe kolieska tak, aby sa robot pohyboval po kružnici s priemerom 10 cm.

Mohli by sme zadať, nech sa pohybuje s nejakou  $v_T$ , ale to sa ťažko meria. Preto skúsime nájsť vhodné hodnoty a nech odmerajú omega – rýchlosť otáčania.

Toto je už trochu náročnejšia úloha, použijeme výsledok odvodený na prednáške, prípadne odvodíme...

TODO: Odvodenie.

Robot sa v prípade nerovnakých rýchlostí koliesok bude stáčať na tú stranu, kde má pomalšie koliesko. Tento pohyb možno charakterizovať ako pohyb po oblúku okolo tzv. *okamžitého streda otáčania* (ICR - Instantaneous Center of Rotation), pričom polomer tohto oblúka označíme  $r$ . Pokiaľ označíme symbolmi  $v_L$  rýchlosť ľavého a  $v_R$  rýchlosť pravého kolesa, pre polomer otáčania okolo ICR platí

$$r = \frac{L}{2} \frac{v_R + v_L}{v_R - v_L} \quad (8.2)$$

Pre okamžitú tangenciálnu rýchlosť streda vozíka T platí:

$$v_T = \frac{v_R + v_L}{2} \quad (8.3)$$

A pre a uhlovú rýchlosť otáčania vozíka okolo ICR platí:

$$\omega_T = \frac{v_R - v_L}{L} \quad (8.4)$$

Ak uvedené vzťahy nebudeme odvodzovať, ale ich prevezmeme z prednášok, môžeme si aspoň overiť ich platnosť na základe toho, čo vieme z predošlých dvoch úloh. Aký je polomer  $r$  otáčania, ak sa obe hnacie kolesá otáčajú rovnakou uhlovou rýchlosťou v rovnakom smere, a ako je to ak sa otáčajú v opačnom smere?

Ak by mali zadanú aj rýchlosť, vyriešia dve rovnice o dvoch neznámych a majú výsledok. Keďže sme rýchlosť nezadali, musia si niečo zvoliť. To úlohu značne sťaží :-)

Rýchlosti sú z intervalu  $< 0, 99 >$ . Skutočné jednotky nás pre priemer nezaujímajú, vykrátia sa. Ale ak vypočítame  $v_T$ , bude zasa len v inžinierskych jednotkách. Jednu rýchlosť treba zvoliť, tak nech teda  $v_R = 99$ , aby sa kružnica kreslila čo najrýchlejšie. Po dosadení vyjde

$$100 = \frac{180}{2} \frac{99 + v_L}{99 - v_L}$$

$$\frac{200}{180}(99 - v_L) = 99 + v_L$$

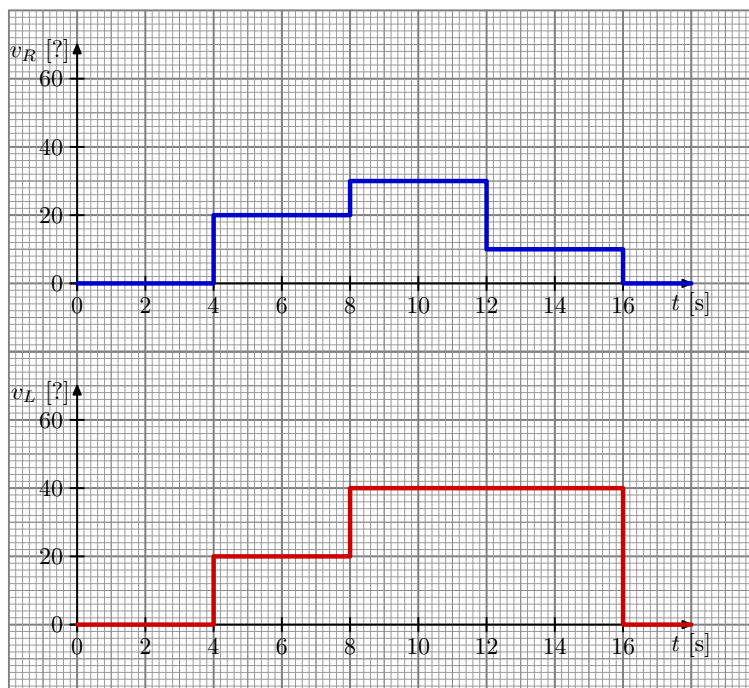
$$v_L = 99 \frac{20.180}{380.180} = \underline{\underline{5,2}}$$

Vypočítané hodnoty vyskúšajte a overte. Odmerajte priemer nakreslenej kružnice (Ako sa meria priemer kružnice?). Dosiahli ste výsledok s uspokojivou presnosťou? Ak nie, čo je na príčine a ako by ste ho zlepšili?

Je zrejmé, že pri použití rýchlosti 5 namiesto vypočítanej 5,2 dostaneme iný rozmer kružnice (99,57 mm !). Ak sa pozrieme dobre na posledný zlomok, vidno, že stačí zmeniť  $v_R$  na 95 a výsledok bude presne  $v_L = 5$ . Potom aj kružnica bude presne 100 mm. Ďalšie vhodné dvojice  $[v_R, v_L]$  sú  $[95, 5]$ ,  $[76, 4]$ ,  $[57, 3]$ ,  $[38, 2]$ ,  $[19, 1]$ . Samozrejme, že sa budú líšiť rýchlosťou pohybu.

## Úloha 4: Kinematický model

Nakreslite trajektóriu Robotnačky pre zadané priebehy rýchlostí koliesok.



Obr. 29: Priebehy rýchlostí koliesok Robotnačky.

Podľa predošlej úlohy by sme vedeli určiť rýchlosť pohybu robota i jeho otáčanie, ale na určenie trajektórie by sme museli prepočítať jednotlivé zložky pohybu do súradnicového systému v ktorom by sme ju chceli zachytiť.

Preto použijeme transformačnú maticu na prevod medzi lokálnymi súradnicami robota do globálnej súradnicovej sústavy.

$$\dot{x} = \frac{R}{2}(v_R + v_L) \cos \theta \quad (8.5)$$

$$\dot{y} = \frac{R}{2}(v_R + v_L) \sin \theta \quad (8.6)$$

$$\dot{\theta} = \frac{R}{L}(v_R - v_L). \quad (8.7)$$

Vyskúšajte:

```
#include "robotdll.h"

int main(int argc, char *argv[])
{
    int r = robot_init(1);           // Initialize
    robot_user(r, "john", "turtle"); // Set User / Password
    robot_connect(r, "com5");        // Open Connection
    robot_alwayswait(r, "off");      // Wait for finished commands

    robot_rtspeed(r, 0, 0);          // Interval 1: 0..4s
    robot_wait(r, 4000);             // Do it for 4 sec

    robot_rtspeed(r, 20, 20);        // Interval 2: 4..8s
    robot_wait(r, 4000);             // Do it for 4 sec
}
```



```
robot_rtspeed(r,40,30);      // Interval 3: 8..12s
robot_wait(r,4000);          // Do it for 4 sec

robot_rtspeed(r,40,10);      // Interval 4: 12..16s
robot_wait(r,4000);          // Do it for 4 sec

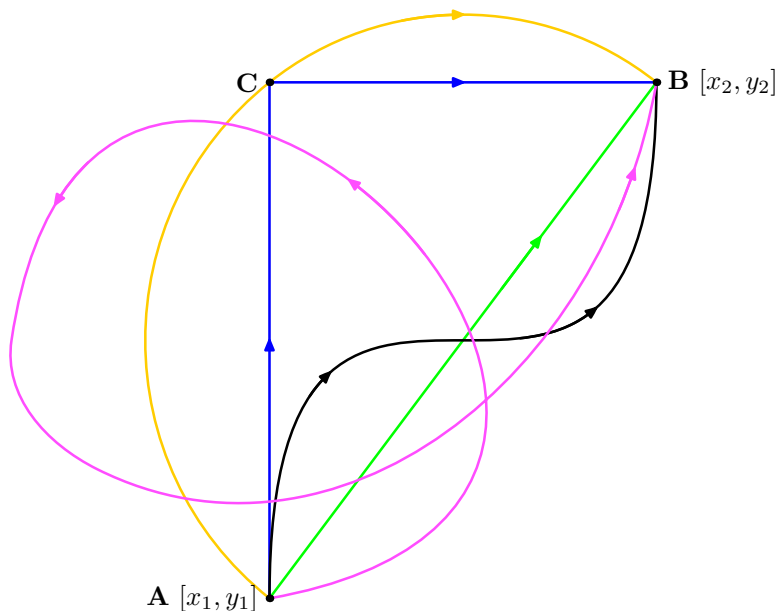
robot_rtspeed(r,0,0);         // Interval 5: 16..
robot_wait(r,4000);          // Do it for 4 sec

robot_done(r);               // Close Connection
return 0;
}
```

### Úloha 5: Z bodu A do bodu B

Napište program, ktorý Robotnačku prevedie z počiatočnej polohy  $[0,0]$  do bodu so súradnicami  $[30\text{cm}, 50\text{cm}]$ .

A je tu ďalšia komplikácia. Zatiaľ čo prvé tri úlohy mali jednoznačné riešenie, táto úloha predstavuje tzv. *inverznú kinematickú úlohu*. Riešení inverznej úlohy je nekonečne veľa, pozrite napr. na tento obrázok



Obr. 30: Možnosti putovania z A do B.

Ani podmienka, že pritom treba prejsť cez daný bod (napr. C na obrázku) nám nepomôže (žltá a modrá cesta). Samozrejme, že si môžeme stanoviť ďalšie obmedzujúce podmienky, ktorými počet možných riešení podstatne obmedzíme. Typicky to býva optimalizácia spotreby energie, prípadne najkratšia možná cesta (nie vždy je to to isté, viete uviesť príklad?). Pre Robotnačku môže byť takým obmedzením napríklad požiadavka, aby sa kolieskom nechodilo po už nakreslených čiarach.

Jeden z možných algoritmov ako vo všeobecnosti riešiť túto úlohu jednoznačne spočíva v tom, že použijeme len dve operácie – OTOČ a POSUŇ. Takže robota najprv nasmerujeme na daný cieľ a potom sa tam len presunie. Prípadne na konci prevedieme ešte jedno otočenie do požadovaného smeru.